

# PATENT APPLICATION FEE DETERMINATION RECORD

Effective October 1, 1994

Application or Docket Number

324,443

## CLAIMS AS FILED - PART I

(Column 1)

(Column 2)

SMALL ENTITY

OR

OTHER THAN  
SMALL ENTITY

FOR	NUMBER FILED	NUMBER EXTRA
BASIC FEE		
TOTAL CLAIMS	43 minus 20 =	* 23
INDEPENDENT CLAIMS	4 minus 3 =	* 1
MULTIPLE DEPENDENT CLAIM PRESENT		

RATE	FEE
	365.00
x\$11=	
x38=	
+120=	
TOTAL	

OR

OR

OR

OR

OR

RATE	FEE
	730.00
x\$22=	506
x76=	76
+240=	
TOTAL	1312

\* If the difference in column 1 is less than zero, enter "0" in column 2

## CLAIMS AS AMENDED - PART II

(Column 1)

(Column 2)

(Column 3)

SMALL ENTITY

OR

OTHER THAN  
SMALL ENTITY

AMENDMENT A	CLAIMS REMAINING AFTER AMENDMENT	HIGHEST NUMBER PREVIOUSLY PAID FOR	PRESENT EXTRA
Total	* 56	Minus ** 43	= 13
Independent	* 8	Minus *** 4	= 4
FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM			

RATE	ADDITIONAL FEE
x\$11=	143
x38=	156
+120=	0
TOTAL ADDIT. FEE	299

OR

OR

OR

OR

RATE	ADDITIONAL FEE
x\$22=	
x76=	
+240=	
TOTAL ADDIT. FEE	

AMENDMENT B	CLAIMS REMAINING AFTER AMENDMENT	HIGHEST NUMBER PREVIOUSLY PAID FOR	PRESENT EXTRA
Total	*	Minus **	=
Independent	*	Minus ***	=
FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM			

RATE	ADDITIONAL FEE
x\$11=	
x38=	
+120=	
TOTAL ADDIT. FEE	

OR

OR

OR

OR

RATE	ADDITIONAL FEE
x\$22=	
x76=	
+240=	
TOTAL ADDIT. FEE	

AMENDMENT C	CLAIMS REMAINING AFTER AMENDMENT	HIGHEST NUMBER PREVIOUSLY PAID FOR	PRESENT EXTRA
Total	*	Minus **	=
Independent	*	Minus ***	=
FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM			

RATE	ADDITIONAL FEE
x\$11=	
x38=	
+120=	
TOTAL ADDIT. FEE	

OR

OR

OR

OR

RATE	ADDITIONAL FEE
x\$22=	
x76=	
+240=	
TOTAL ADDIT. FEE	

\* If the entry in column 1 is less than the entry in column 2, write "0" in column 3.  
 \*\* If the "Highest Number Previously Paid For" IN THIS SPACE is less than 20, enter "20."  
 \*\*\* If the Highest Number Previously Paid For IN THIS SPACE is less than 3, enter "3."  
 The Highest Number Previously Paid For (Total or Independent) is the highest number found in the appropriate box in column 1.

# PACE DATA ENTRY CODING SHEET

# 1ST EXAM.

## 2ND EXAMINER

**APPLICATION NUMBER**

TYPE  
APPL

FILING DATE

**SPECIAL  
HANDLING**

## GROUP ART UNIT

18/32443

1	0	1	9	4
---	---	---	---	---



2	3	0	1
---	---	---	---

**TOTAL  
CLAIMS**

## INDEPENDENT CLAIMS

## SMALL ENTITY?

**FILING FEE**

**FOREIGN  
LICENSE**

**ATTORNEY DOG**

3	7	—
---	---	---

4			
---	--	--	--

1	4	4	2
---	---	---	---



023075

## CONTINUITY DATA

CONT	STATUS
CODE	CODE

**PARENT APPLICATION  
SERIAL NUMBER**

**PCT APPLICATION SERIAL NUMBER**

**PAREN**  
**NU:**

[illegible]

## PCT/FOREIGN APPLICATION DATA

**FOREIGN  
PRIORITY  
CLAIMED**

COUNTRY  
CODE

PCT/FOREIGN APPLICATION SERIAL NUMBER

1	2	3	4	5
---	---	---	---	---


[illegible]



## PATENT APPLICATION

EMBEDDED PROGRAM OBJECTS IN  
DISTRIBUTED HYPERMEDIA SYSTEMS

## Inventors:

Michael Doyle,  
David Martin,  
Cheong Ang

## Assignee:

University of California  
1320 Harbor Bay Parkway, Suite 150  
Alameda, CA 94502

TOWNSEND and TOWNSEND KHOURIE and CREW  
Steuart Street Tower, 20th Floor  
One Market Plaza  
San Francisco, California 94105  
(415) 543-9600

10/2/94 10:23:00

08/324443

TOWNSEND and TOWNSEND KHOURIE and CREW  
Steuart Street Tower  
One Market Plaza  
San Francisco, CA 94105  
(415) 543-9600

Atty. Docket No. 02307-553

"Express Mail" Label No. TB380892941US

Date of Deposit October 17, 1994

PATENT APPLICATION  
COMMISSIONER OF PATENT AND TRADEMARKS  
Washington, D. C. 20231



Sir:

Transmitted herewith for filing is the

- ☒ patent application of  
☐ design patent application of  
☐ continuation-in-part patent application of

Inventors: Michael Doyle, David Martin and Cheong Ang

I hereby certify that this is being deposited with the United States Postal Service "Express Mail Post Office to Addressee" service under 37 CFR 1.10 on the date indicated above and is addressed to the Commissioner of Patents and Trademarks, Washington, D. C. 20231

By

Sunil Dutt  
Sunil Dutt

For: EMBEDDED PROGRAM OBJECTS IN DISTRIBUTED HYPERMEDIA SYSTEMS

Enclosed are:

- ☒ 10 sheets of ☐ formal ☒ informal drawings.  
☐ An assignment of the invention to \_\_\_\_\_  
☐ A ☐ signed ☐ unsigned Declaration & Power of Attorney.  
☒ A ☐ signed ☒ unsigned Declaration.  
☐ A Power of Attorney.  
☐ A verified statement to establish small entity status under 37 CFR 1.9 and 37 CFR 1.27.  
☐ A certified copy of a \_\_\_\_\_ application.  
☐ Information Disclosure Statement under 37 CFR 1.97.  
☒ Appendix A; Appendix B

In view of the Unsigned Declaration as filed with this application and pursuant to 37 CFR §1.53(d), Applicant requests deferral of the filing fee until submission of the Missing Parts of Application.

DO NOT CHARGE THE FILING FEE AT THIS TIME.

Telephone:  
(415) 543-9600

APPNOFEE.TRN 12/92

Charles J. Kulas  
Charles J. Kulas  
Reg. No.: 35,809  
Attorneys for Applicant

PATENT

Attorney Docket No. 2307U-553

EMBEDDED PROGRAM OBJECTS IN  
DISTRIBUTED HYPERMEDIA SYSTEMSNotice Regarding Copyrighted Material

A portion of the disclosure of this patent document contains material which is subject to copyright protection. The copyright owner has no objection to the facsimile reproduction by anyone of the patent document or the patent disclosure as it appears in the Patent and Trademark Office file or records, but otherwise reserves all copyright rights whatsoever.

## BACKGROUND OF THE INVENTION

This invention relates generally to manipulating data in a computer network, and specifically to retrieving, presenting and manipulating embedded program objects in distributed hypermedia systems.

Computer networks are becoming increasingly popular as a medium for locating and accessing a wide range of data from locations all over the world. The most popular global network is the Internet with millions of computer systems connected to it. The Internet has become popular due to widely adopted standard protocols that allow a vast interconnection of computers and localized computer networks to communicate with each other. Computer systems connected to a network such as the Internet may be of varying types, e.g., mainframes, workstations, personal computers, etc. The computers are manufactured by different companies using proprietary hardware and operating systems and thus have incompatibilities in their instruction sets, busses, software, file formats and other aspects of their architecture and operating systems. Localized computer networks connected to the Internet may be incompatible with other computer systems and localized networks in terms of the physical layer of communication including the specific hardware used to

implement the network. Also, different networks use differing, incompatible protocols for transferring information and are not able to communicate with each other without a translation mechanism such as a "gateway".

5           The Internet provides a uniform and open standard for allowing various computers and networks to communicate with each other. For example, the Internet uses Transfer Control Protocol/Internet Protocol ("TCP/IP") that defines a uniform packet-switched communication standard which is  
10 ultimately used in every transfer of information that takes place over the Internet.

Other Internet standards are the HyperText Transmission Protocol ("HTTP") that allows hypertext documents to be exchanged freely among any computers connected to the  
15 Internet and HyperText Markup Language ("HTML") that defines the way in which hypertext documents designate links to information. See, e.g., Berners-Lee, T. J., "The world-wide web," Computer Networks and ISDN Systems 25 (1992).

20           A hypertext document is a document that allows a user to view a text document displayed on a display device connected to the user's computer and to access, retrieve and view other data objects that are linked to hypertext words or phrases in the hypertext document. In a hypertext document, the user may "click on," or select, certain words or phrases  
25 in the text that specify a link to other documents, or data objects. In this way, the user is able to navigate easily among data objects. The data objects may be local to the user's computer system or remotely located over a network. An early hypertext system is Hypercard, by Apple Computer, Inc.  
30 Hypercard is a standalone system where the data objects are local to the user's system.

When a user selects a phrase in a hypertext document that has an associated link to another document, the linked document is retrieved and displayed on the user's display  
35 screen. This allows the user to obtain more information in an efficient and easy manner. This provides the user with a simple, intuitive and powerful way to "branch off" from a main document to learn more about topics of interest.

Objects may be text, images, sound files, video data, documents or other types of information that is presentable to a user of a computer system. When a document is primarily text and includes links to other data objects according to the hypertext format, the document is said to be a hypertext document. When graphics, sound, video or other media capable of being manipulated and presented in a computer system is used as the object linked to, the document is said to be a hypermedia document. A hypermedia document is similar to a hypertext document, except that the user is able to click on images, sound icons, video icons, etc., that link to other objects of various media types, such as additional graphics, sound, video, text, or hypermedia or hypertext documents.

Fig. 1 shows examples of hypertext and hypermedia documents and links associating data objects in the documents to other data objects. Hypermedia document 10 includes hypertext 20, an image icon at 22, a sound icon at 24 and more hypertext 26. Fig. 1 shows hypermedia document 10 substantially as it would appear on a user's display screen. The user is able to select, or "click" on icons and text on a display screen by using an input device, such as a mouse, in a manner well-known in the art.

When the user clicks on the phrase "hypermedia," software running on the user's computer obtains the link associated with the phrase, symbolically shown by arrow 30, to access hypermedia document 14. Hypermedia document 14 is retrieved and displayed on the user's display screen. Thus, the user is presented with more information on the phrase "hypermedia." The mechanism for specifying and locating a linked object such as hypermedia document 14 is an HTML "element" that includes an object address in the format of a Uniform Resource Locator (URL).

Similarly, additional hypertext 26 can be selected by the user to access hypertext document 12 via link 32 as shown in Fig. 1. If the user selects additional hypertext 26, then the text for hypertext document 12 is displayed on the user screen. Note that hypertext document 12, itself, has hypertext at 28. Thus, the user can click on the phrase

15  
20  
25  
30  
35

4

"hypermedia" while viewing document 12 to access hypermedia document 14 in a manner similar to that discussed above.

Documents, and other data objects, can be referenced by many links from many different source documents. Fig. 1 shows document 14 serving as a target link for both documents 10 and 12. A distributed hypertext or hypermedia document typically has many links within it that specify many different data objects located in computers at different geographical locations connected by a network. Hypermedia document 10 includes image icon 22 with a link to image 16. One method of viewing images is to include an icon, or other indicator, within the text.

Typically, the indicator is a very small image and may be a scaled down version of the full image. The indicator may be shown embedded within the text when the text is displayed on the display screen. The user may select the indicator to obtain the full image. When the user clicks on image icon 22 browser software executing on the user's computer system retrieves the corresponding full image, e.g., a bit map, and displays it by using external software called a "viewer." This results in the full image, represented by image 16, being displayed on the screen.

An example of a browser program is the National Center for Supercomputing Application's (NCSA) Mosaic software developed by the University of Illinois at Urbana/Champaign, Illinois. Another example is "Cello" available on the Internet at <http://www.law.cornell.edu/>. Many viewers exist that handle various file formats such as ".TIF," ".GIF," formats. When a browser program invokes a viewer program, the viewer is launched as a separate process. The view displays the full image in a separate "window" (in a windowing environment) or on a separate screen. This means that the browser program is no longer active while the viewer is active. By using indicators to act as place holders for full images that are retrieved and displayed only when a user selects the indicator, data traffic over the network is reduced. Also, since the retrieval and display of large images may require several seconds or more of transfer time



the user does not have to wait to have images transferred that are of no interest to the user.

Returning to Fig. 1, another type of data object is a sound object shown as sound icon 24 within the hypermedia document. When the user selects sound icon 24, the user's computer accesses sound data shown symbolically by data file 40. The accessed sound data plays through a speaker or other audio device.

As discussed above, hypermedia documents allow a user to access different data objects. The objects may be text, images, sound files, video, additional documents, etc. As used in this specification, a data object is information capable of being retrieved and presented to a user of a computer system. Some data objects include executable code combined with data. An example of such a combination is a "self-extracting" data object that includes code to "unpack" or decompress data that has been compressed to make it smaller before transferring. When a browser retrieves an object such as a self-extracting data object the browser may allow the user to "launch" the self-extracting data object to automatically execute the unpacking instructions to expand the data object to its original size. Such a combination of executable code and data is limited in that the user can do no more than invoke the code to perform a singular function such as performing the self-extraction after which time the object is a standard data object.

Other existing approaches to embedding interactive program objects in documents include the Object Linking and Embedding (OLE) facility in Microsoft Windows, by Microsoft Corp., and OpenDoc, by Apple Computer, Inc. At least one shortcoming of these approaches is that neither is capable of allowing a user to access embedded interactive program objects in distributed hypermedia documents over networks.

Fig. 2 is an example of a computer network. In Fig. 2, computer systems are connected to Internet 100, although in practice Internet 100 may be replaced by any suitable computer network. In Fig. 2, a user 102 operates a small computer 104, such as a personal computer or a work station. The user's

6

computer is equipped with various components, such as user input devices (mouse, trackball, keyboard, etc.), a display device (monitor, liquid crystal display (LCD), etc.), local storage (hard disk drive, etc.), and other components.

5 Typically, small computer 104 is connected to a larger computer, such as server A at 106. The larger computer may have additional users and computer systems connected to it, such as computer 108 operated by user 110. Any group of computers may form a localized network. A localized network  
10 does not necessarily adopt the uniform protocols of the larger interconnecting network (i.e., Internet 100) and is more geographically constrained than the larger network. The localized network may connect to the larger network through a "gateway" or "node" implemented on, for example, a server.

15 Internet 100 connects other localized networks, such as server B at 120, which interconnects users 122, 124 and 126 and their respective computer systems to Internet 100.

Internet 100 is made up of many interconnected computer systems and communication links. Communication links may be  
20 by hardwire, fiber optic cable, satellite or other radio wave propagation, etc. Data may move from server A to server B through any number of intermediate servers and communication links or other computers and data processing equipment not shown in Fig. 2 but symbolically represented by Internet 100.

25 A user at a workstation or personal computer need not connect to the Internet via a larger computer, such as server A or server B. This is shown, for example, by small computer 130 connected directly to Internet 100 as by a telephone modem or other link. Also, a server need not have  
30 users connected to it locally, as is shown by server C at 132 of Fig. 2. Many configurations of large and small computers are possible.

Typically, a computer on the Internet is characterized as either a "client" or "server" depending on  
35 the role that the computer is playing with respect to requesting information or providing information. Client computers are computers that typically request information from a server computer which provides the information. For

this reason, servers are usually larger and faster machines that have access to many data files, programs, etc., in a large storage associated with the server. However, the role of a server may also be adopted by a smaller machine depending on the transaction. That is, user 110 may request information via their computer 108 from server A. At a later time, server A may make a request for information from computer 108. In the first case, where computer 108 issues a request for information from server A, computer 108 is a "client" making a request of information from server A. Server A may have the information in a storage device that is local to Server A or server A may have to make requests of other computer systems to obtain the information. User 110 may also request information via their computer 108 from a server, such as server B located at a remote geographical location on the Internet. However, user 110 may also request information from a computer, such as small computer 124, thus placing small computer 124 in the role of a "server." For purposes of this specification, client and server computers are categorized in terms of their predominant role as either an information requestor or provider. Clients are generally information requestors, while servers are generally information providers.

Referring again to Fig. 1, data objects such as distributed hypermedia documents 10, 12 and 14, image 16 and sound data file 40, may be located at any of the computers shown in Fig. 2. Since these data objects may be linked to a document located on another computer the Internet allows for remote object linking.

For example, hypertext document 10 of Fig. 1 may be located at user 110's client computer 108. When user 110 makes a request by, for example, clicking on hypertext 20 (i.e., the phrase "hypermedia"), user 110's small client computer 108 processes links within hypertext document 10 to retrieve document 14. In this example, we assume that document 14 is stored at a remote location on server B. Thus, in this example, computer 108 issues a command that includes the address of document 14. This command is routed through server A and Internet 100 and eventually is received by server

B. Server B processes the command and locates document 14 on its local storage. Server 14 then transfers a copy of the document back to client 108 via Internet 100 and server A. After client computer 108 receives document 14, it is displayed so that user 110 may view it.

Similarly, image object 16 and sound data file 40 may reside at any of the computers shown in Fig. 2. Assuming image object 16 resides on server C when user 110 clicks on image icon 22, client computer 108 generates a command to retrieve image object 16 to server C. Server C receives the command and transfers a copy of image object 16 to client computer 108. Alternatively, an object, such as sound data file 40, may reside on server A so that it is not necessary to traverse long distances via the Internet in order to retrieve the data object.

The Internet is said to provide an "open distributed hypermedia system." It is an "open" system since Internet 100 implements a standard protocol that each of the connecting computer systems, 106, 130, 120, 132 and 134 must implement (TCP/IP). It is a "hypermedia" system because it is able to handle hypermedia documents as described above via standards such as the HTTP and HTML hypertext transmission and mark up standards, respectively. Further, it is a "distributed" system because data objects that are imbedded within a document may be located on many of the computer systems connected to the Internet. An example of an open distributed hypermedia system is the so-called "world-wide web" implemented on the Internet and discussed in papers such as the Berners-Lee reference given above.

The open distributed hypermedia system provided by the Internet allows users to easily access and retrieve different data objects located in remote geographic locations on the Internet. However, this open distributed hypermedia system as it currently exists has shortcomings in that today's large data objects are limited largely by bandwidth constraints in the various communication links in the Internet and localized networks, and by the limited processing power, or computing constraints, of small computer systems normally

100-101-102-103-104-105-106-107-108-109-110-111-112-113-114-115-116-117-118-119-120-121-122-123-124-125-126-127-128-129-130-131-132-133-134-135-136-137-138-139-140-141-142-143-144-145-146-147-148-149-150-151-152-153-154-155-156-157-158-159-160-161-162-163-164-165-166-167-168-169-170-171-172-173-174-175-176-177-178-179-180-181-182-183-184-185-186-187-188-189-190-191-192-193-194-195-196-197-198-199-200-201-202-203-204-205-206-207-208-209-210-211-212-213-214-215-216-217-218-219-220-221-222-223-224-225-226-227-228-229-230-231-232-233-234-235-236-237-238-239-240-241-242-243-244-245-246-247-248-249-250-251-252-253-254-255-256-257-258-259-260-261-262-263-264-265-266-267-268-269-270-271-272-273-274-275-276-277-278-279-280-281-282-283-284-285-286-287-288-289-290-291-292-293-294-295-296-297-298-299-300-301-302-303-304-305-306-307-308-309-310-311-312-313-314-315-316-317-318-319-320-321-322-323-324-325-326-327-328-329-330-331-332-333-334-335-336-337-338-339-340-341-342-343-344-345-346-347-348-349-350-351-352-353-354-355-356-357-358-359-360-361-362-363-364-365-366-367-368-369-370-371-372-373-374-375-376-377-378-379-380-381-382-383-384-385-386-387-388-389-390-391-392-393-394-395-396-397-398-399-400-401-402-403-404-405-406-407-408-409-410-411-412-413-414-415-416-417-418-419-420-421-422-423-424-425-426-427-428-429-430-431-432-433-434-435-436-437-438-439-440-441-442-443-444-445-446-447-448-449-450-451-452-453-454-455-456-457-458-459-460-461-462-463-464-465-466-467-468-469-470-471-472-473-474-475-476-477-478-479-480-481-482-483-484-485-486-487-488-489-490-491-492-493-494-495-496-497-498-499-500-501-502-503-504-505-506-507-508-509-510-511-512-513-514-515-516-517-518-519-520-521-522-523-524-525-526-527-528-529-530-531-532-533-534-535-536-537-538-539-540-541-542-543-544-545-546-547-548-549-550-551-552-553-554-555-556-557-558-559-560-561-562-563-564-565-566-567-568-569-570-571-572-573-574-575-576-577-578-579-580-581-582-583-584-585-586-587-588-589-590-591-592-593-594-595-596-597-598-599-600-601-602-603-604-605-606-607-608-609-610-611-612-613-614-615-616-617-618-619-620-621-622-623-624-625-626-627-628-629-630-631-632-633-634-635-636-637-638-639-640-641-642-643-644-645-646-647-648-649-650-651-652-653-654-655-656-657-658-659-660-661-662-663-664-665-666-667-668-669-670-671-672-673-674-675-676-677-678-679-680-681-682-683-684-685-686-687-688-689-690-691-692-693-694-695-696-697-698-699-700-701-702-703-704-705-706-707-708-709-710-711-712-713-714-715-716-717-718-719-720-721-722-723-724-725-726-727-728-729-730-731-732-733-734-735-736-737-738-739-740-741-742-743-744-745-746-747-748-749-750-751-752-753-754-755-756-757-758-759-760-761-762-763-764-765-766-767-768-769-770-771-772-773-774-775-776-777-778-779-780-781-782-783-784-785-786-787-788-789-790-791-792-793-794-795-796-797-798-799-800-801-802-803-804-805-806-807-808-809-810-811-812-813-814-815-816-817-818-819-820-821-822-823-824-825-826-827-828-829-830-831-832-833-834-835-836-837-838-839-840-841-842-843-844-845-846-847-848-849-850-851-852-853-854-855-856-857-858-859-860-861-862-863-864-865-866-867-868-869-870-871-872-873-874-875-876-877-878-879-880-881-882-883-884-885-886-887-888-889-890-891-892-893-894-895-896-897-898-899-900-901-902-903-904-905-906-907-908-909-910-911-912-913-914-915-916-917-918-919-920-921-922-923-924-925-926-927-928-929-930-931-932-933-934-935-936-937-938-939-940-941-942-943-944-945-946-947-948-949-950-951-952-953-954-955-956-957-958-959-960-961-962-963-964-965-966-967-968-969-970-971-972-973-974-975-976-977-978-979-980-981-982-983-984-985-986-987-988-989-990-991-992-993-994-995-996-997-998-999-1000

35

30

5

10

15

25

9

provided to most users. Large data objects are difficult to update at frame rates fast enough (e.g., 30 frames per second) to achieve smooth animation. Moreover, the processing power needed to perform the calculations to animate such images in real time does not exist on most workstations, not to mention personal computers. Today's browsers and viewers are not capable of performing the computation necessary to generate and render new views of these large data objects in real time.

For example, the Internet's open distributed hypermedia system allows users to view still images. These images are simple non-interactive two-dimensional images, similar to photographs. Much digital data available today exists in the form of high-resolution multi-dimensional image data (e.g., three dimensional images) which is viewed on a computer while allowing the user to perform real time viewing transformations on the data in order for the user to better understand the data.

An example of such type of data is in medical imaging where advanced scanning devices, such as Magnetic Resonance Imaging (MRI) and Computed Tomography (CT), are widely used in the fields of medicine, quality assurance and meteorology to present physicians, technicians and meteorologists with large amounts of data in an efficient way. Because visualization of the data is the best way for a user to grasp the data's meaning, a variety of visualization techniques and real time computer graphics methods have been developed. However, these systems are bandwidth-intensive and compute-intensive and often require multiprocessor arrays and other specialized graphics hardware to carry them out in real time. Also, large amounts of secondary storage for data are required. The expense of these requirements has limited the ability of researchers to readily exchange findings since these larger computers required to store, present and manipulate images are not available to many of the researchers that need to have access to the data.

On the other hand, small client computers in the form of personal computers or workstations such as client computer 108 of Fig. 2 are generally available to a much

10  
15  
20  
25  
30  
35

10

larger number of researchers. Further, it is common for these smaller computers to be connected to the Internet. Thus, it is desirable to have a system that allows the accessing, display and manipulation of large amounts of data, especially image data, over the Internet to a small, and relatively cheap, client computer.

Due to the relatively low bandwidth of the Internet (as compared to today's large data objects) and the relatively small amount of processing power available at client computers, many valuable tasks performed by computers cannot be performed by users at client computers on the Internet. Also, while the present open distributed hypermedia system on the Internet allows users to locate and retrieve data objects it allows users very little, if any, interaction with these data objects. Users are limited to traditional hypertext and hypermedia forms of selecting linked data objects for retrieval and launching viewers or other forms of external software to have the data objects presented in a comprehensible way.

Thus, it is desirable to have a system that allows a user at a small client computer connected to the Internet to locate, retrieve and manipulate data objects when the data objects are bandwidth-intensive and compute-intensive. Further, it is desirable to allow a user to manipulate data objects in an interactive way to provide the user with a better understanding of information presented and to allow the user to accomplish a wider variety of tasks.

#### SUMMARY OF THE INVENTION

The present invention provides a method for running embedded program objects in a computer network environment. The method includes the steps of providing at least one client workstation and one network server coupled to the network environment where the network environment is a distributed hypermedia environment; displaying, on the client workstation, a portion of a hypermedia document received over the network from the server, where the hypermedia document includes an embedded controllable application; and interactively

controlling the embedded controllable application from the client workstation via communication sent over the distributed hypermedia environment.

The present invention allows a user at a client computer connected to a network to locate, retrieve and manipulate objects in an interactive way. The invention not only allows the user to use a hypermedia format to locate and retrieve program objects, but also allows the user to interact with an application program located at a remote computer.

Interprocess communication between the hypermedia browser and the embedded application program is ongoing after the program object has been launched. The user is able to use a vast amount of computing power beyond that which is contained in the user's client computer.

In one application, high resolution three dimensional images are processed in a distributed manner by several computers located remotely from the user's client computer. This amounts to providing parallel distributed processing for tasks such as volume rendering or three dimensional image transformation and display. Also, the user is able to rotate, scale and otherwise reposition the viewpoint with respect to these images without exiting the hypermedia browser software. The control and interaction of viewing the image may be provided within the same window that the browser is using assuming the environment is a "windowing" environment. The viewing transformation and volume rendering calculations may be performed by remote distributed computer systems.

Once an image representing a new viewpoint is computed the frame image is transmitted over the network to the user's client computer where it is displayed at a designated position within a hypermedia document. By transmitting only enough information to update the image, the need for a high bandwidth data connection is reduced. Compression can be used to further reduce the bandwidth requirements for data transmission.

Other applications of the invention are possible. For example, the user can operate a spreadsheet program that

102 FOR ATTACHED

12

is being executed by one or more other computer systems connected via the network to the user's client computer. Once the spreadsheet program has calculated results, the results may be sent over the network to the user's client computer for display to the user. In this way, computer systems located remotely on the network can be used to provide the computing power that may be required for certain tasks and to reduce the data bandwidth by only transmitting results of the computations.

Still other applications of the present invention are possible, as disclosed in the specification, below.

#### BRIEF DESCRIPTION OF THE DRAWINGS

Fig. 1 illustrates examples of hypertext and hypermedia documents and links;

Fig. 2 is an example of a computer network;

Fig. 3 is an illustration of a computer system suitable for use with the present invention;

Fig. 4 is an illustration of basic subsystems in the computer system of Fig. 3;

Fig. 5 is an illustration of an embodiment of the invention using a client computer, server computer and a network;

Fig. 6 shows another embodiment of the present invention using additional computers on the network;

Fig. 7A is a flowchart of some of the functionality within the HTMLparse.c file;

Fig. 7B is a flowchart of some of the functionality within the HTMLformat.c file;

Fig. 8A is a flowchart of some of the functionality within the HTMLwidget.c file;

Fig. 8B is a flowchart of some of the functionality within the HTML.c file;

Fig. 9 is a screen display generated in accordance with the present invention; and

Fig. 10 is a diagram of the various processes and data paths in the present invention.

102 FOR C-ATTACH

13



5

10

15

25

35

4

Additional subsystems such as a printer, keyboard, fixed disk and others are shown. Peripherals and input/output (I/O) devices can be connected to the computer system by, for example serial port 185. For example, serial port 185 can be used to connect the computer system to a modem for connection to a network or serial port 185 can be used to interface with a mouse input device. The interconnection via system bus 184 allows central processor 180 to communicate with each subsystem and to control the execution of instructions from system memory 181 or fixed disk 186, and the exchange of information between subsystems. Other arrangements of subsystems and interconnections are possible.

Fig. 5 is an illustration of an embodiment of the invention using a client computer, server computer and a network.

In Fig. 5, client computer 200 communicates with server computer 204 via network 206. Both client computer 200 and server computer 204 use a network protocol layer to communicate with network 206. In a preferred embodiment, network 206 is the Internet and the network protocol layers are TCP/IP. Other networks and network protocols may be used. For ease of illustration, additional hardware and software layers are not shown in Fig. 5.

Client computer 200 includes processes, such as browser client 208 and application client 210. In a preferred embodiment, application client 210 is resident within client computer 200 prior to browser client 208's parsing of a hypermedia document as discussed below. In a preferred embodiment application client 210 resides on the hard disk or RAM of client computer 200 and is loaded (if necessary) and executed when browser client 208 detects a link to application client 210. The preferred embodiment uses the XEvent interprocess communication protocol to exchange information between browser client 208 and application client 210 as described in more detail, below. Another possibility is to install application client 210 as a "terminate and stay resident" (TSR) program in an operating system environment,

1024 FOR "STANDARD"

15

such as X-Window. Thereby making access to application client 210 much faster.

Browser client 208 is a process that a user of client computer 200 invokes in order to access various data objects, such as hypermedia documents, on network 206. Hypermedia document 212 shown within client computer 200 is an example of a hypermedia document, or object, that a user has requested access to. In this example, hypermedia document 212 has been retrieved from a server connected to network 206 and has been loaded into, e.g., client computer 200's RAM or other storage device.

Once hypermedia document 212 has been loaded into client computer 200, browser client 208 parses hypermedia document 212. In parsing hypermedia document 212, browser client 208 detects links to data objects as discussed above in the Background of the Invention section. In Fig. 5, hypermedia document 212 includes an embedded program link at 214. Embedded program link 214 identifies application client 212 as an application to invoke. In this present example, the application, namely, application client 210, resides on the same computer as the browser client 208 that the user is executing to view the hypermedia document. Embedded program link 214 may include additional information, such as parameters, that tell application client 210 how to proceed. For example, embedded program link 214 may include a specification as to a data object that application client 210 is to retrieve and process.

When browser client 208 encounters embedded program link 214, it invokes application client 210 (optionally, with parameters or other information) and application client 210 executes instructions to perform processing in accordance with the present invention.

An example of the type of processing that application client 210 may perform is multidimensional image visualization. Note that application client 210 is in communication with network 206 via the network protocol layer of client computer 200. This means that application client 210 can make requests over network 206 for data objects, such

162 FOR CATTAGE

16

as multidimensional image objects. For example, application client 210 may request an object, such as object 1 at 216, located in server computer 204. Application client 210 may make the request by any suitable means. Assuming network 206 is the Internet, such a request would typically be made by using HTTP in response to a HTML-style link definition for embedded program link 214.

Assuming application client 210 has made a request for the data object at 216, server process 218 ultimately receives the request. Server process 218 then retrieves data object 216 and transfers it over network 206 back to application client 210. To continue with the example of a multidimensional visualization application, data object 216 may be a three dimensional view of medical data for, e.g., an embryo.

After application client 210 receives the multidimensional data object 216, application client 210 executes instructions to display the multidimensional embryo data on the display screen to a user of the client computer 200. The user is then able to interactively operate controls to recompute different views for the image data. In a preferred embodiment, a control window is displayed within, or adjacent to, a window generated by browser client 208 that contains a display of hypermedia document 212. An example of such display is discussed below in connection with Fig. 9. Thus, the user is able to interactively manipulate a multidimensional image object by means of the present invention. In order to make application client 210 integral with displays created by browser client 208, both the browser client and the application client must be in communication with each other, as shown by the arrow connecting the two within client computer 200. The manner of communication is through an application program interface (API), discussed below.

Browser client 208 is a process, such as NCSA Mosaic, Cello, etc. Application client 210 is embodied in software presently under development called "VIS" and "Panel" created by the Center for Knowledge Management at the

17

University of California, San Francisco, as part of the Doyle Group's distributed hypermedia object embedding approach described in "Integrated Control of Distributed Volume Visualization Through the World-Wide-Web," by C. Ang, D. Martin, M. Doyle; to be published in the Proceedings of Visualization 1994, IEEE Press, Washington, D.C., October 1994.

Versions and descriptions of software embodying the present invention are generally available as hyperlinked data objects from the Visible Embryo Project's World Wide Web document at the URL address "HTTP://visembryo.ucsf.edu/".

Another embodiment of the present invention uses an application server process executing on server computer 204 to assist in processing that may need to be performed by an external program. For example, in Fig. 5, application server 220 resides on server computer 204. Application server 220 works in communication with application client 210 residing on client computer 200. In a preferred embodiment, application server 220 is called VRServer, also a part of Doyle Group's approach. Since server computer 204 is typically a larger computer having more data processing capabilities and larger storage capacity, application server 220 can operate more efficiently, and much faster, than application client 210 in executing complicated and numerous instructions.

In the present example where a multidimensional image object representing medical data for an embryo is being viewed, application server 220 could perform much of the viewing transformation and volume rendering calculations to allow a user to interactively view the embryo data at their client computer display screen. In a preferred embodiment, application client 210 receives signals from a user input device at the user's client computer 200. An example of such input would be to rotate the embryo image from a current position to a new position from the user's point of view. This information is received by application client 210 and processed to generate a command sent over network 206 to application server 220. Once application server 220 receives the information in the form of, e.g., a coordinate

102-107-111-112-113-114-115-116-117-118-119-120-121-122-123-124-125-126-127-128-129-130-131-132-133-134-135-136-137-138-139-140-141-142-143-144-145-146-147-148-149-150-151-152-153-154-155-156-157-158-159-160-161-162-163-164-165-166-167-168-169-170-171-172-173-174-175-176-177-178-179-180-181-182-183-184-185-186-187-188-189-190-191-192-193-194-195-196-197-198-199-200-201-202-203-204-205-206-207-208-209-210-211-212-213-214-215-216-217-218-219-220-221-222-223-224-225-226-227-228-229-230-231-232-233-234-235-236-237-238-239-240-241-242-243-244-245-246-247-248-249-250-251-252-253-254-255-256-257-258-259-260-261-262-263-264-265-266-267-268-269-270-271-272-273-274-275-276-277-278-279-280-281-282-283-284-285-286-287-288-289-290-291-292-293-294-295-296-297-298-299-300-301-302-303-304-305-306-307-308-309-310-311-312-313-314-315-316-317-318-319-320-321-322-323-324-325-326-327-328-329-330-331-332-333-334-335-336-337-338-339-340-341-342-343-344-345-346-347-348-349-350-351-352-353-354-355-356-357-358-359-360-361-362-363-364-365-366-367-368-369-370-371-372-373-374-375-376-377-378-379-380-381-382-383-384-385-386-387-388-389-390-391-392-393-394-395-396-397-398-399-400-401-402-403-404-405-406-407-408-409-410-411-412-413-414-415-416-417-418-419-420-421-422-423-424-425-426-427-428-429-430-431-432-433-434-435-436-437-438-439-440-441-442-443-444-445-446-447-448-449-450-451-452-453-454-455-456-457-458-459-460-461-462-463-464-465-466-467-468-469-470-471-472-473-474-475-476-477-478-479-480-481-482-483-484-485-486-487-488-489-490-491-492-493-494-495-496-497-498-499-500-501-502-503-504-505-506-507-508-509-510-511-512-513-514-515-516-517-518-519-520-521-522-523-524-525-526-527-528-529-530-531-532-533-534-535-536-537-538-539-540-541-542-543-544-545-546-547-548-549-550-551-552-553-554-555-556-557-558-559-560-561-562-563-564-565-566-567-568-569-570-571-572-573-574-575-576-577-578-579-580-581-582-583-584-585-586-587-588-589-590-591-592-593-594-595-596-597-598-599-600-601-602-603-604-605-606-607-608-609-610-611-612-613-614-615-616-617-618-619-620-621-622-623-624-625-626-627-628-629-630-631-632-633-634-635-636-637-638-639-640-641-642-643-644-645-646-647-648-649-650-651-652-653-654-655-656-657-658-659-660-661-662-663-664-665-666-667-668-669-670-671-672-673-674-675-676-677-678-679-680-681-682-683-684-685-686-687-688-689-690-691-692-693-694-695-696-697-698-699-700-701-702-703-704-705-706-707-708-709-710-711-712-713-714-715-716-717-718-719-720-721-722-723-724-725-726-727-728-729-730-731-732-733-734-735-736-737-738-739-740-741-742-743-744-745-746-747-748-749-750-751-752-753-754-755-756-757-758-759-760-761-762-763-764-765-766-767-768-769-770-771-772-773-774-775-776-777-778-779-780-781-782-783-784-785-786-787-788-789-790-791-792-793-794-795-796-797-798-799-800-801-802-803-804-805-806-807-808-809-810-811-812-813-814-815-816-817-818-819-820-821-822-823-824-825-826-827-828-829-830-831-832-833-834-835-836-837-838-839-840-841-842-843-844-845-846-847-848-849-850-851-852-853-854-855-856-857-858-859-860-861-862-863-864-865-866-867-868-869-870-871-872-873-874-875-876-877-878-879-880-881-882-883-884-885-886-887-888-889-890-891-892-893-894-895-896-897-898-899-900-901-902-903-904-905-906-907-908-909-910-911-912-913-914-915-916-917-918-919-920-921-922-923-924-925-926-927-928-929-930-931-932-933-934-935-936-937-938-939-940-941-942-943-944-945-946-947-948-949-950-951-952-953-954-955-956-957-958-959-960-961-962-963-964-965-966-967-968-969-970-971-972-973-974-975-976-977-978-979-980-981-982-983-984-985-986-987-988-989-990-991-992-993-994-995-996-997-998-999-1000-1001-1002-1003-1004-1005-1006-1007-1008-1009-1010-1011-1012-1013-1014-1015-1016-1017-1018-1019-1020-1021-1022-1023-1024-1025-1026-1027-1028-1029-1030-1031-1032-1033-1034-1035-1036-1037-1038-1039-1040-1041-1042-1043-1044-1045-1046-1047-1048-1049-1050-1051-1052-1053-1054-1055-1056-1057-1058-1059-1060-1061-1062-1063-1064-1065-1066-1067-1068-1069-1070-1071-1072-1073-1074-1075-1076-1077-1078-1079-1080-1081-1082-1083-1084-1085-1086-1087-1088-1089-1090-1091-1092-1093-1094-1095-1096-1097-1098-1099-1100-1101-1102-1103-1104-1105-1106-1107-1108-1109-1110-1111-1112-1113-1114-1115-1116-1117-1118-1119-1120-1121-1122-1123-1124-1125-1126-1127-1128-1129-1130-1131-1132-1133-1134-1135-1136-1137-1138-1139-1140-1141-1142-1143-1144-1145-1146-1147-1148-1149-1150-1151-1152-1153-1154-1155-1156-1157-1158-1159-1160-1161-1162-1163-1164-1165-1166-1167-1168-1169-1170-1171-1172-1173-1174-1175-1176-1177-1178-1179-1180-1181-1182-1183-1184-1185-1186-1187-1188-1189-1190-1191-1192-1193-1194-1195-1196-1197-1198-1199-1200-1201-1202-1203-1204-1205-1206-1207-1208-1209-1210-1211-1212-1213-1214-1215-1216-1217-1218-1219-1220-1221-1222-1223-1224-1225-1226-1227-1228-1229-1230-1231-1232-1233-1234-1235-1236-1237-1238-1239-1240-1241-1242-1243-1244-1245-1246-1247-1248-1249-1250-1251-1252-1253-1254-1255-1256-1257-1258-1259-1260-1261-1262-1263-1264-1265-1266-1267-1268-1269-1270-1271-1272-1273-1274-1275-1276-1277-1278-1279-1280-1281-1282-1283-1284-1285-1286-1287-1288-1289-1290-1291-1292-1293-1294-1295-1296-1297-1298-1299-1300-1301-1302-1303-1304-1305-1306-1307-1308-1309-1310-1311-1312-1313-1314-1315-1316-1317-1318-1319-1320-1321-1322-1323-1324-1325-1326-1327-1328-1329-1330-1331-1332-1333-1334-1335-1336-1337-1338-1339-1340-1341-1342-1343-1344-1345-1346-1347-1348-1349-1350-1351-1352-1353-1354-1355-1356-1357-1358-1359-1360-1361-1362-1363-1364-1365-1366-1367-1368-1369-1370-1371-1372-1373-1374-1375-1376-1377-1378-1379-1380-1381-1382-1383-1384-1385-1386-1387-1388-1389-1390-1391-1392-1393-1394-1395-1396-1397-1398-1399-1400-1401-1402-1403-1404-1405-1406-1407-1408-1409-1410-1411-1412-1413-1414-1415-1416-1417-1418-1419-1420-1421-1422-1423-1424-1425-1426-1427-1428-1429-1430-1431-1432-1433-1434-1435-1436-1437-1438-1439-1440-1441-1442-1443-1444-1445-1446-1447-1448-1449-1450-1451-1452-1453-1454-1455-1456-1457-1458-1459-1460-1461-1462-1463-1464-1465-1466-1467-1468-1469-1470-1471-1472-1473-1474-1475-1476-1477-1478-1479-1480-1481-1482-1483-1484-1485-1486-1487-1488-1489-1490-1491-1492-1493-1494-1495-1496-1497-1498-1499-1500-1501-1502-1503-1504-1505-1506-1507-1508-1509-1510-1511-1512-1513-1514-1515-1516-1517-1518-1519-1520-1521-1522-1523-1524-1525-1526-1527-1528-1529-1530-1531-1532-1533-1534-1535-1536-1537-1538-1539-1540-1541-1542-1543-1544-1545-1546-1547-1548-1549-1550-1551-1552-1553-1554-1555-1556-1557-1558-1559-1560-1561-1562-1563-1564-1565-1566-1567-1568-1569-1570-1571-1572-1573-1574-1575-1576-1577-1578-1579-1580-1581-1582-1583-1584-1585-1586-1587-1588-1589-1590-1591-1592-1593-1594-1595-1596-1597-1598-1599-1600-1601-1602-1603-1604-1605-1606-1607-1608-1609-1610-1611-1612-1613-1614-1615-1616-1617-1618-1619-1620-1621-1622-1623-1624-1625-1626-1627-1628-1629-1630-1631-1632-1633-1634-1635-1636-1637-1638-1639-1640-1641-1642-1643-1644-1645-1646-1647-1648-1649-1650-1651-1652-1653-1654-1655-1656-1657-1658-1659-1660-1661-1662-1663-1664-1665-1666-1667-1668-1669-1670-1671-1672-1673-1674-1675-1676-1677-1678-1679-1680-1681-1682-1683-1684-1685-1686-1687-1688-1689-1690-1691-1692-1693-1694-1695-1696-1697-1698-1699-1700-1701-1702-1703-1704-1705-1706-1707-1708-1709-1710-1711-1712-1713-1714-1715-1716-1717-1718-1719-1720-1721-1722-1723-1724-1725-1726-1727-1728-1729-1730-1731-1732-1733-1734-1735-1736-1737-1738-1739-1740-1741-1742-1743-1744-1745-1746-1747-1748-1749-1750-1751-1752-1753-1754-1755-1756-1757-1758-1759-1760-1761-1762-1763-1764-1765-1766-1767-1768-1769-1770-1771-1772-1773-1774-1775-1776-1777-1778-1779-1780-1781-1782-1783-1784-1785-1786-1787-1788-1789-1790-1791-1792-1793-1794-1795-1796-1797-1798-1799-1800-1801-1802-1803-1804-1805-1806-1807-1808-1809-1810-1811-1812-1813-1814-1815-1816-1817-1818-1819-1820-1821-1822-1823-1824-1825-1826-1827-1828-1829-1830-1831-1832-1833-1834-1835-1836-1837-1838-1839-1840-1841-1842-1843-1844-1845-1846-1847-1848-1849-1850-1851-1852-1853-1854-1855-1856-1857-1858-1859-1860-1861-1862-1863-1864-1865-1866-1867-1868-1869-1870-1871-1872-1873-1874-1875-1876-1877-1878-1879-1880-1881-1882-1883-1884-1885-1886-1887-1888-1889-1890-1891-1892-1893-1894-1895-1896-1897-1898-1899-1900-1901-1902-1903-1904-1905-1906-1907-1908-1909-1910-1911-1912-1913-1914-1915-1916-1917-1918-1919-1920-1921-1922-1923-1924-1925-1926-1927-1928-1929-1930-1931-1932-1933-1934-1935-1936-1937-1938-1939-1940-1941-1942-1943-1944-1945-1946-1947-1948-1949-1950-1951-1952-1953-1954-1955-1956-1957-1958-1959-1960-1961-1962-1963-1964-1965-1966-1967-1968-1969-1970-1971-1972-1973-1974-1975-1976-1977-1978-1979-1980-1981-1982-1983-1984-1985-1986-1987-1988-1989-1990-1991-1992-1993-1994-1995-1996-1997-1998-1999-2000-2001-2002-2003-2004-2005-2006-2007-2008-2009-2010-2011-2012-2013-2014-2015-2016-2017-2018-2019-2020-2021-2022-2023-2024-2025-2026-2027-2028-2029-2030-2031-2032-2033-2034-2035-2036-2037-2038-2039-2040-2041-2042-2043-2044-2045-2046-2047-2048-2049-2050-2051-2052-2053-2054-2055-2056-2057-2058-2059-2060-2061-2062-2063-2064-2065-2066-2067-2068-2069-2070-2071-2072-2073-2074-2075-2076-2077-2078-2079-2080-2081-2082-2083-2084-2085-2086-2087-2088-2089-2090-2091-2092-2093-2094-2095-2096-2097-2098-2099-2100-2101-2102-2103-2104-2105-2106-2107-2108-2109-2110-2111-2112-2113-2114-2115-2116-2117-2118-2119-2120-2121-2122-2123-2124-2125-2126-2127-2128-2129-2130-2131-2132-2133-2134-2135-2136-2137-2138-2139-2140-2141-2142-2143-2144-2145-2146-2147-2148-2149-2150-2151-2152-2153-2154-2155-2156-2157-2158-2159-2160-2161-2162-2163-2164-2165-2166-2167-2168-2169-2170-2171-2172-2173-2174-2175-2176-2177-2178-2179-2180-2181-2182-2183-2184-2185-2186-2187-2188-2189-2190-2191-2192-2193-2194-2195-2196-2197-2198-2199-2200-2201-2202-2203-2204-2205-2206-2207-2208-2209-2210-2211-2212-2213-2214-2215-2216-2217-2218-2219-2220-2221-2222-2223-2224-2225-2226-2227-2228-2229-2230-2231-2232-2233-2234-2235-2236-2237-2238-2239-2240-2241-2242-2243-2244-2245-2246-2247-2248-2249-2250-2251-2252-2253-2254-2255-2256-2257-2258-2259-2260-2261-2262-2263-2264-2265-2266-2267-2268-2269-2270-2271-2272-2273-2274-2275-2276-2277-2278-2279-2280-2281-2282-2283-2284-2285-2286-2287-2288-2289-2290-2291-2292-2293-2294-2295-2296-2297-2298-2299-2300-2301-2302-2303-2304-2305-2306-2307-2308-2309-2310-2311-2312-2313-2314-2315-2316-2317-2318-2319-2320-2321-2322-2323-2324-2325-2326-2327-2328-2329-2330-2331-2332-2333-2334-2335-2336-2337-2338-2339-2340-2341-2342-2343-2344-2345-2346-2347-2348-2349-2350-2351-2352-2353-2354-2355-2356-2357-2358-2359-2360-2361-2362-2363-2364-2365-2366-2367-2368-2369-2370-2371-2372-2373-2374-2375-2376-2377-2378-2379-2380-2381-2382-2383-2384-2385-2386-2387-2388-2389-2390-2391-2392-2393-2394-2395-2396-2397-2398-2399-2400-2401-2402-2403-2404-2405-2406-2407-2408-2409-2410-2411-2412-2413-2414-2415-2416-2417-2418-2419-2420-2421-2422-2423-2424-2425-2426-2427-2428-2429-2430-2431-2432-2433-2434-2435-2436-2437-2438-2439-2440-2441-2442-2443-2444-2445-2446-2447-2448-2449-2450-2451-2452-2453-2454-2455-2456-2457-2458-2459-2460-2461-2462-2463-2464-2465-2466-2467-2468-2469-2470-2471-2472-2473-2474-2475-2476-2477-2478-2479-2480-2481-2482-2483-2484-2485-2486-2487-2488-2489-2490-2491-2492-2493-2494-2495-2496-2497-2498-2499-2500-2501-2502-2503-2504-2505-2506-2507-2508-2509-2510-2511-2512-2513-2514-2515-2516-2517-2518-2519-2520-2521-2522-2523-2524-2525-2526-2527-2528-2529-2530-2531-2532-2533-2534-2535-2536-2537-2538-2539-2540-2541-2542-2543-2544-2545-2546-2547-2548-2549-2550-2551-2552-2553-2554-2555-2556-2557-2558-2559-2560-2561-2562-2563-2564-2565-2566-2567-2568-2569-2570-2571-2572-2573-2574-2575-2576-2577-2578-2579-2580-2581-2582-2583-2584-2585-2586-2587-2588-2589-2590-2591-2592-2593-2594-2595-2596-2597-2598-2599-2600-2601-2602-2603-2604-2605-2606-2607-2608-2609-2610-2611-2612-2613-2614-2615-2616-2617-2618-2619-2620-2621-2622-2623-2624-2625-2626-2627-2628-2629-2630-2631-2632-2633

transformation for a new viewing position, application server 220 performs the mathematical calculations to compute a new view for the embryo image. Once the new view has been computed, the image data for the new view is sent over network 206 to application client 210 so that application client 210 can update the viewing window currently displaying the embryo image. In a preferred embodiment, application server 220 computes a frame buffer of raster display data, e.g., pixel values, and transfers this frame buffer to application client 210. Techniques, such as data compression and delta encoding, can be used to compress the data before transmitting over network 206 to reduce the bandwidth requirement.

It will be readily seen that application server 220 can advantageously use server computer 204's computing resources to perform the viewing transformation much more quickly than could application client 210 executing on client computer 200. Further, by only transmitting the updated frame buffer containing a new view for the embryo image, the amount of data sent over network 206 is reduced. By using appropriate compression techniques, such as, e.g., MPEG (Motion Picture Experts Group) or JPEG (Joint Photographic Experts Group), efficient use of network 206 is preserved.

Fig. 6 shows yet another embodiment of the present invention. Fig. 6 is similar to Fig. 5, except that additional computers 222 and 224 are illustrated. Each additional computer includes a process labeled "Application (Distributed)." The distributed application performs a portion of the task that an application, such as application server 220 or application client 210, perform. In the present example, tasks such as volume rendering may be broken up and easily performed among two or more computers. These computers can be remote from each other on network 206. Thus, several computers, such as server computer 204 and additional computers 222 and 224 can all work together to perform the task of computing a new viewpoint and frame buffer for the embryo for the new orientation of the embryo image in the present example. The coordination of the distributed processing can be performed at client computer 200 by

application client 210, at server computer 204 by application server 220, or by any of the distributed applications executing on additional computers, such as 222 and 224. In a preferred embodiment, distributed processing is coordinated by a program called "VIS" represented by application client 210 in Fig. 6.

Other applications of the invention are possible. For example, the user can operate a spreadsheet program that is being executed by one or more other computer systems connected via the network to the user's client computer. Once the spreadsheet program has calculated results, those results may be sent over the network to the user's client computer for display within the hypermedia document on the user's client computer. In this way, computer systems located remotely on the network can be used to provide the computing power that may be required for certain tasks and to reduce the data bandwidth required by only transmitting results of the computations.

Another type of possible application of this invention would involve embedding a program which runs only on the client machine, but which provides the user with more functionality than exists in the hypermedia browser alone. An example of this is an embedded client application which is capable of viewing and interacting with images which have been processed with Dr. Doyle's MetaMAP invention (US Patent 4,847,604). This MetaMAP process uses object-oriented color map processing to allow individual color index ranges within paletted images to have object identities, and is useful for the creation of, for example, interactive picture atlases. It is a more efficient means for defining irregular "hotspots" on images than the ISMAP function of the World Wide Web, which uses polygonal outlines to define objects in images. A MetaMAP-capable client-based image browser application can be embedded, together with an associated image, within a hypermedia document, allowing objects within the MetaMAP-processed image to have URL addresses associated with them. When a user clicks with a mouse upon an object within the MetaMAP-processed image, the MetaMAP client application

relays the relevant URL back to the hypermedia browser application, which then retrieves the HTML file or hypermedia object which corresponds to that URL.

The various processes in the system of the present invention communicate through a custom API called Mosaic/External Application Program Interface MEAPI. The MEAPI set of predefined messages includes those shown in Table I.

10	Message Function	Message Name
	-----	-----
	Messages from server to client:	
	1. Server Update Done	XtNrefreshNotify
	2. Server Ready	XtNpanelStartNotify
15	3. Server Exiting	XtNpanelExitNotify
	Messages from client to server:	
	4. Area Shown	XtNmapNotify
	5. Area Hidden	XtNunmapNotify
	6. Area Destroyed	XtNexitNotify

Table I

The messages in Table I are defined in the file protocol\_lib.h in Appendix B. The functions of the MEAPI are provided in protocol\_lib.c of Appendix B. Thus, by using MEAPI a server process communicates to a client application program to let the client application know when the server has finished updating information, such as an image frame buffer, or pixmap (Message 1); when the server is ready to start processing messages (Message 2) and when the server is exiting or stopping computation related to the server application program.

For client to server communications, MEAPI provides for the client informing the server when the image display window area is visible, when the area is hidden and when the area is destroyed. Such information allows the server to decide whether to allocate computing resources for, e.g., rendering and viewing transformation tasks where the server is running an application program to generate new views of a multi dimensional object. Source code for MEAPI fundamental functions such as handle\_client\_msg, register\_client, register\_client\_msg\_callback and send\_client\_msg may be found in protocol\_lib.c as part of the source code in Appendix B.



Next, a discussion of the software processes that perform parsing of a hypermedia document and launching of an application program is provided in connection with Table II and Figs. 7A, 7B, 8A and 8B.

Table II, below, shows an example of an HTML tag format used by the present invention to embed a link to an application program within a hypermedia document.

```
<EMBED
  TYPE = "type"
  HREF = "href"
  WIDTH = width
  HEIGHT = height
>
```

TABLE II

As shown in Table II, the EMBED tag includes TYPE, HREF, WIDTH and HEIGHT elements. The TYPE element is a Multipurpose Internet Mail Extensions (MIME) type. Examples of values for the TYPE element are "application/x-vis" or "video/mpeg". The type "application /x-vis" indicates that an application named "x-vis" is to be used to handle the object at the URL specified by the HREF. Other types are possible such as "application/x-inventor", "application/postscript" etc. In the case where TYPE is "application/x-vis" this means that the object at the URL address is a three dimensional image object since the program "x-vis" is a data visualization tool designed to operate on three dimensional image objects. However, any manner of application program may be specified by the TYPE element so that other types of applications, such as a spreadsheet program, database program, word processor, etc. may be used with the present invention. Accordingly, the object reference by the HREF element would be, respectively, a spreadsheet object, database object, word processor document object, etc.

On the other hand, TYPE values such as "video/mpeg", "image/gif", "video/x-sgi-movie", etc. describe the type of data that HREF specifies. This is useful where an external application program, such as a video player, needs to know what format the data is in, or where the browser client needs

to determine which application to launch based on the data format. Thus, the TYPE value can specify either an application program or a data type. Other TYPE values are possible. HREF specifies a URL address as discussed above for a data object. Where TYPE is "application/x-vis" the URL address specifies a multi-dimensional image object. Where TYPE is "video/mpeg" the URL address specifies a video object.

WIDTH and HEIGHT elements specify the width and height dimensions, respectively, of a Distributed Hypermedia Object Embedding (DHOE) window to display an external application object such as the three dimensional image object or video object discussed above.

Fig. 7A is a flowchart describing some of the functionality within the HTMLparse.c file of routines. The routines in HTMLparse.c perform the task of parsing a hypermedia document and detecting the EMBED tag. In a preferred embodiment, the enhancements to include the EMBED tag are made to an HTML library included in public domain NCSA Mosaic version 2.4. ~~These files are included as source code in Appendix A attached to this specification.~~ Note that much of the source code in ~~Appendix A~~ is pre-existing NCSA Mosaic code. Only those portions of the source code that relate to the new functionality discussed in this specification should be considered as part of the invention. The new functionality is identifiable as being set off from the main body of source code by conditional compilation macros such as "#ifdef ... #endif" as will be readily apparent to one of skill in the art.

In general, the flowcharts in this specification illustrate one or more software routines executing in a computer system such as computer system 1 of Fig. 1. The routines may be implemented by any means as is known in the art. For example, any number of computer programming languages, such as "C", Pascal, FORTRAN, assembly language, etc., may be used. Further, various programming approaches such as procedural, object oriented or artificial intelligence techniques may be employed.

The steps of the flowcharts may be implemented by one or more software routines, processes, subroutines, modules, etc. It will be apparent that each flowchart is illustrative of merely the broad logical flow of the method of the present invention and that steps may be added to, or taken away from, the flowcharts without departing from the scope of the invention. Further, the order of execution of steps in the flowcharts may be changed without departing from the scope of the invention. Additional considerations in implementing the method described by the flowchart in software may dictate changes in the selection and order of steps. Some considerations are event handling by interrupt driven, polled, or other schemes. A multiprocessing or multitasking environment could allow steps to be executed "concurrently." For ease of discussion the implementation of each flowchart may be referred to as if implemented in a single "routine".

The modifications to NCSA Mosaic version 2.4 software files HTMLparse.c, HTMLformat.c, HTMLwidget.c and HTML.c will next be discussed, in turn.

Returning to Fig. 7, it is assumed that a hypermedia document has been obtained at a user's client computer and that a browser program executing on the client computer displays the document and calls a first routine in the HTMLparse.c file called "HTMLparse". This first routine, HTMLparse, is entered at step 252 where a pointer to the start of the document portion is passed. Steps 254, 256 and 258 represent a loop where the document is parsed or scanned for HTML tags or other symbols. While the file HTMLparse.c includes routines to handle all possible tags and symbols that may be encountered, Fig. 7A, for simplicity, only illustrates the handling of EMBED tags.

Assuming there is more text to parse, execution proceeds to step 256 where routines in HTMLparse.c obtain the next item (e.g., word, tag or symbol) from the document. At step 258 a check is made as to whether the current tag is the EMBED tag. If not, execution returns to step 254 where the next tag in the document is obtained. If, at step 258, it is determined that the tag is the EMBED tag, execution proceeds

to step 260 where an enumerated type is assigned for the tag. Each occurrence of a valid EMBED tag specifies an embedded object. HTMLParse calls a routine "get\_mark" in HTMLparse.c to put sections of HTML document text into a "markup" text data structure. Routine get\_mark, in turn, calls ParseMarkType to assign an enumerated type. The enumerated type is an identifier with a unique integer associated with it that is used in later processing described below.

Once all of the hypermedia text in the text portion to be displayed has been parsed, execution of HTMLparse.c routines terminates at step 262.

Fig. 7B is a flowchart of routines in file HTMLformat.c to process the enumerated type created for the EMBED tag by routines in HTMLparse.c. In the X-Window implementation of a preferred embodiment, the enumerated type is processed as if it is a regular Motif/XT widget. For details on X-Window development see, e.g., "Xlib Programming Manual," "X Toolkit Intrinsics Programming Manual" and "Motif Programming Manual" published by O'Reilly & Associates, Inc. HTMLformat is entered at step 270 where a pointer to the enumerated type to process is passed.

At step 272 the parameters of the structure are initialized in preparation for inserting a DrawingArea widget on an HTML page. This includes providing values for the width and height of a window on the display to contain an image, position of the window, style, URL of the image object, etc. Various codes are also added by routines in HTMLformat.c (such as TriggerMarkChanges) to insert an internal representation of the HTML statement into an object list maintained internally by the browser. In the X-Window application corresponding to the source code of Appendix A, the browser is NCSA Mosaic version 2.4.

Fig. 8A is a flowchart for routine HTMLwidget. HTMLwidget creates display data structures and launches an external application program to handle the data object specified by the URL in the EMBED tag.

HTMLwidget is entered at step 280 after HTMLformat has created the internal object representation of the EMBED

25

tag. HTMLwidget is passed the internal object and performs its processing on the object. At step 282 the DrawingArea widget is created according to the type of the internal representation, from HTMLformat, specified in the internal object. Similarly, at step 284 a pixmap area for backing storage is defined.

At step 286 a check is made as to whether the type attribute of the object, i.e., the value for the TYPE element of the EMBED tag, is an application. If so, step 290 is executed to launch a predetermined application. In a preferred embodiment an application is launched according to a user-defined list of application type/application pairs. The list is defined as a user-configurable XResource as described in "Xlib Programming Manual." An alternative embodiment could use the MIME database as the source of the list of application type/application pairs. The routine "vis\_start\_external\_application" in file HTMLformat.c is invoked to match the application type and to identify the application to launch.

The external application is started as a child process of the current running process (Mosaic), and informed about the window ID of the DrawingArea created in HTMLformat. The external application is also passed information about the ID of the pixmap, the data URL and dimensions. Codes for communication such as popping-up/iconifying, start notification, quit notification and refresh notification with external applications and DrawingArea refreshing are also added. Examples of such codes are (1) "setup/start" in vis\_register\_client and vis\_get\_panel\_window in HTMLwidgets.c; (2) "handle messages from external applications" in vis\_handle\_panel\_msg in HTMLwidgets.c; (3) "send messages to external applications" in vis\_send\_msg in HTMLwidgets.c; (4) "terminate external applications" in vis\_exit in HTMLwidgets.c which calls vis\_send\_msg to send a quit message; and (5) "respond to refresh msgs" in vis\_redraw in HTMLwidgets.c.

If, at step 286, the type is determined not to be an application object (e.g., a three dimensional image object in the case of application "x-vis") a check is made at step 288

to determine if the type is a video object. If so, step 292 is executed to launch a video player application. Parameters are passed to the video player application to allow the player to display the video object within the DrawingArea within the display of the portion of hypermedia document on the client's computer. Note that many other application objects types are possible as described above.

Fig. 8B is a flowchart for routine HTML. Routine HTML takes care of "shutting down" the objects, data areas, etc. that were set up to launch the external application and display the data object. HTML is entered at step 300 and is called when the display or other processing of the EMBED tag has been completed. At step 302 the display window is removed and the memory areas for the pixmap and internal object structure is made free for other uses. Completion of processing can be by user command or by computer control.

The present invention allows a user to have interactive control over application objects such as three dimensional image objects and video objects. In a preferred embodiment, controls are provided on the external applications' user interface. In the case of a VIS/panel application, a process, "panel" creates a graphical user interface (GUI) thru which the user interacts with the data. The application program, VIS, can be executing locally with the user's computer or remotely on a server, or on one or more different computers, on the network. The application program updates pixmap data and transfers the pixmap data (frame image data) to a buffer to which the browser has access. The browser only needs to respond to the refresh request to copy the contents from the updated pixmap to the DrawingArea. The Panel process sends messages as "Msg" sending performed by routines such as vis\_send\_msg and vis\_handle\_panel\_msg to send events (mousemove, keypress, etc.) to the external application.

Fig. 9 is a screen display of the invention showing an interactive application object (in this case a three dimensional image object) in a window within a browser window. In Fig. 9, the browser is NCSA Mosaic version 2.4. The

processes VIS, Panel and VRServer work as discussed above.

Fig. 9 shows screen display 356 Mosaic window 350 containing image window 352 and a portion of a panel window 354. Note

that image window 352 is within Mosaic window 350 while panel

window 354 is external to Mosaic window 350. Another

possibility is to have panel window 354 within Mosaic window

350. By using the controls in panel window 354 the user is

able to manipulate the image within image window 352 in real

time do perform such operations as scaling, rotation,

translation, color map selection, etc. In Fig. 9, two Mosaic

windows are being used to show two different views of an

embryo image. One of the views is rotated by six degrees from

the other view so that a stereoscopic effect can be achieved

when viewing the images. Communication between Panel and VIS

is via "Tooltalk" described in, e.g., "Tooltalk 1.1.1

Reference Manual," from SunSoft.

Fig. 10 is an illustration of the processes VIS,

Panel and VRServer discussed above. As shown in Fig. 10, the

browser process, Mosaic, communicates with the Panel process

via inter-client communication mechanisms such as provided in

the X-Window environment. The Panel process communicates with

the VIS process through a communications protocol (ToolTalk,

in the preferred embodiment) to exchange visualization command

messages and image data. The image data is computed by one or

more copies of a process called VRServer that may be executing

on remote computers on the network. VRServer processes

respond to requests such as rendering requests to generate

image segments. The image segments are sent to VIS and

combined into a pixmap, or frame image, by VIS. The frame

image is then transferred to the Mosaic screen via

communications between VIS, Panel and Mosaic. A further

description of the data transfer may be found in the paper

"Integrated Control of Distributed Volume Visualization

Through the World-Wide-Web," referenced above.

In the foregoing specification, the invention has

been described with reference to a specific exemplary

embodiment thereof. It will, however, be evident that various

modifications and changes may be made thereunto without

464101-27113630

28

departing from the broader spirit and scope of the invention as set forth in the appended claims. For example, various programming languages and techniques can be used to implement the disclosed invention. Also, the specific logic presented to accomplish tasks within the present invention may be modified without departing from the scope of the invention. Many such changes or modifications will be readily apparent to one of ordinary skill in the art. The specification and drawings are, accordingly, to be regarded in an illustrative rather than a restrictive sense, the invention being limited only by the provided claims.

USPTO 00000000

29



WHAT IS CLAIMED IS:

- 1           1.    A method for running an application program in  
2   a computer network environment, comprising:  
3           providing at least one client workstation and one  
4   network server coupled to said network environment, wherein  
5   said network environment is a distributed hypermedia  
6   environment;  
7           displaying, on said client workstation, at least a  
8   portion of a hypermedia document received over said network  
9   from said server, wherein said hypermedia document includes an  
10   embedded controllable application; and  
11           interactively controlling said embedded controllable  
12   application from said client workstation via communications  
13   sent over said distributed hypermedia environment.
- 14           2.   The method of claim 1, wherein the step of  
15   displaying is performed by using a hypermedia browser  
16   application.
- 17           3.   The method of claim 2, wherein instructions for  
18   controlling said embedded controllable application reside on  
19   said network server, wherein said step of interactively  
20   controlling said embedded controllable application includes  
21   the following substeps:  
22           issuing, from the client workstation, one or more  
23   commands to the network server;  
24           executing, on the network server, one or more  
25   instructions in response to said commands;  
26           sending information from said network server to said  
27   client workstation in response to said executed instructions;  
28   and  
29           processing said information at the client  
30   workstation to interactively control said embedded  
31   controllable application.

32 4. The method of claim 2, wherein instructions for  
33 controlling said embedded controllable application reside on  
34 said client workstation.

35 5. The method of claim 2, wherein the  
36 communications to interactively control said embedded  
37 controllable application from said client workstation continue  
38 to be exchanged between the controllable application and the  
39 hypermedia browser even after the controllable application  
40 program has been launched.

41 6. The method of claim 3, wherein said embedded  
42 controllable application is a multi-dimensional viewer.

43 7. The method of claim 3, wherein said embedded  
44 controllable application is a spreadsheet program.

45 8. The method of claim 3, wherein said embedded  
46 controllable application is a database program.

47 9. The method of claim 3, wherein said embedded  
48 controllable application is a word processor.

49 10. The method of claim 3, wherein said substeps of  
50 issuing and sending are via an open protocol.

51 11. The method of claim 10, wherein said open  
52 protocol is an International Standards Organization (ISO)  
53 protocol.

54 12. The method of claim 11, wherein said ISO  
55 protocol is Transfer Control Protocol/Internet Protocol  
56 (TCP/IP) and said network is the Internet.

57 13. The method of claim 12, wherein HyperText  
58 Transfer Protocol is used to transfer said hypermedia document  
59 between said client workstation and said server.

14. The method of claim 13, wherein HyperText Markup Language is used to specify said embedded controllable application within said hypermedia document.

15. A method for running an application program in a computer network environment, comprising:

providing at least one client workstation and one network server coupled to said network environment, said network including a plurality of general purpose workstations, wherein said network environment is a distributed hypermedia environment;

displaying, on said client workstation, at least a portion of a hypermedia document received over said network from said server, wherein said hypermedia document includes at least a first embedded multi-dimensional data visualization application; and

interactively controlling said embedded multi-dimensional data visualization application from said client workstation via communications sent over said distributed hypermedia environment wherein data image rendering is performed by said plurality of general purpose workstations using distributed processing.

16. The method of claim 15, wherein the step of displaying is performed by using a hypermedia browser application.

17. The method of claim 15, wherein the multi-dimensional data visualization includes volume visualization.

18. The method of claim 15, wherein the multi-dimensional data visualization includes two dimensional image processing.

19. The method of claim 15, wherein the multi-dimensional data visualization includes image analysis.

91                   20. The method of claim 15, wherein the multi-  
92 dimensional data visualization includes the display of  
93 animated sequences.

94                   21. The method of claim 15, wherein the multi-  
95 dimensional data visualization includes a geometric data  
96 viewer to display computer aided design files.

97                   22. The method of claim 15, wherein the multi-  
98 dimensional data visualization includes displaying molecular  
99 modeling data.

100                   23. The method of claim 15, wherein a hypermedia  
101 browser is executing on the client workstation, wherein  
102 communications to interactively control said embedded  
103 controllable application from said client workstation continue  
104 to be exchanged between the controllable application and the  
105 hypermedia browser even after the controllable application  
106 program has been launched.

107                   24. A method for interactively controlling an  
108 embedded object in a document displayed on a client computer,  
109 wherein the client computer includes a processor coupled to a  
110 display device and to a user input device, wherein the  
111 processor is further coupled to a computer network, wherein  
112 the computer network is coupled to a server computer and one  
113 or more additional computers, wherein the server computer  
114 includes a local storage device containing a document, wherein  
115 the document includes an embedded object, wherein an  
116 application program for manipulating the embedded object  
117 resides on a first additional computer, the method comprising  
118 the following steps:

119                   transferring, over the network, at least a portion  
120 of the document from the server computer to the client  
121 computer;

122                   accepting first signals from the user input device  
123 that indicate that the embedded object is to be manipulated;

124 issuing commands from the client computer to the  
125 first additional computer in response to the first signals;  
126 executing, by using the first additional computer,  
127 instructions in the application program in response to the  
128 issued commands, wherein the executed instructions generate  
129 information about manipulating the embedded object;  
130 communicating, via the network, the information  
131 about manipulating the embedded object from the first  
132 additional computer to the client computer; and  
133 using the client computer to manipulate the embedded  
134 object according to the communicated information.

135 25. The method of claim 24, wherein said document  
136 is a hypermedia document.

137 26. The method of claim 24, further comprising the  
138 steps of executing instructions in a second application  
139 program on a second additional computer in response to the  
140 issued commands, wherein the instructions executed by the  
141 second additional computer result in information about  
142 manipulating the embedded object being generated more quickly.

143 27. The method of claim 26, wherein said document  
144 is a hypermedia document.

145 28. The method of claim 26, wherein the embedded  
146 object is a multi-dimensional image displayable in any of a  
147 plurality of orientations.

148 29. The method of claim 28, wherein said document  
149 is a hypermedia document.

150 30. The method of claim 28, wherein the executed  
151 instructions perform three dimensional display transformations  
152 to determine the second orientation of the multi-dimensional  
153 image object.

154 31. The method of claim 30, wherein said document  
155 is a hypermedia document.

156 32. The method of claim 28, wherein the executed  
157 instructions perform image rendering to determine an  
158 orientation of the multi-dimensional image.

159 33. The method of claim 32, wherein said document  
160 is a hypermedia document.

161 34. A method for displaying a three dimensional  
162 image object on a client computer, wherein the client computer  
163 includes a processor coupled to a display device, wherein the  
164 processor is further coupled to a computer network, wherein  
165 the computer network is coupled to a server computer and one  
166 or more additional computers, wherein the server computer  
167 includes a local storage device containing a hypermedia  
168 document, wherein the hypermedia document includes a three  
169 dimensional image object embedded within the hypermedia  
170 document, wherein the three dimensional image object is  
171 displayable in a plurality of orientations, the method  
172 comprising the following steps:

173 transferring, over the network, at least a portion  
174 of the hypermedia document from the server computer to the  
175 client computer;

176 displaying on the display device, by using the  
177 processor, at least a portion of the hypermedia document,  
178 wherein the displayed portion of the hypermedia document  
179 includes the three dimensional image object displayed in a  
180 first orientation;

181 using the client computer to issue commands over the  
182 network;

183 executing instruction on a first additional computer  
184 in response to the issued commands, wherein the executed  
185 instructions determine a second orientation for display of the  
186 three dimensional image object;

187           communicating, via the network, information about  
188   the second orientation from the first additional computer to  
189   the client computer; and  
190           using the client computer to redisplay the three  
191   dimensional image object in the second orientation.

192           35. The method of claim 34, wherein said network is  
193   a distributed hypermedia environment.

194           36. The method of claim 34, further comprising the  
195   steps of executing instructions on a second additional  
196   computer in response to the issued commands, wherein the  
197   instructions executed by the second computer enable the second  
198   orientation to be determined more quickly than when only the  
199   first additional computer executes instructions.

200           37. The method of claim 36, wherein said network is  
201   a distributed hypermedia environment.

202           38. The method of claim 36, wherein the executed  
203   instructions perform volume rendering to determine the second  
204   orientation of the three dimensional image object.

205           39. The method of claim 38, wherein said network is  
206   a distributed hypermedia environment.

207           40. The method of claim 36, wherein the executed  
208   instructions perform three dimensional display transformations  
209   to determine the second orientation of the three dimensional  
210   image object.

211           41. The method of claim 40, wherein said network is  
212   a distributed hypermedia environment.

213           42. The method of claim 34, wherein the client  
214   computer includes a user input device coupled to the  
215   processor, the method further comprising the following steps:

216            accepting signals from the user input device,  
217    wherein the accepted signals indicate that the second  
218    orientation is to be determined.

219            43. The method of claim 42, wherein said  
220    network is a distributed hypermedia environment.

10/2/03 4:44 PM

*add*  
*Q7*



## EMBEDDED PROGRAM OBJECTS IN DISTRIBUTED HYPERMEDIA SYSTEMS

### ABSTRACT OF THE DISCLOSURE

5 A system allowing a user of a browser program on a  
computer connected to an open distributed hypermedia system to  
access and execute an embedded program object. The program  
object is embedded into a hypermedia document much like data  
10 objects. The user may select the program object from the  
screen. Once selected the program object executes on the  
user's (client) computer or may execute on a remote server or  
additional remote computers in a distributed processing  
arrangement. After launching the program object, the user is  
15 able to interact with the object as the invention provides for  
ongoing interprocess communication between the application  
object (program) and the browser program. One application of  
the embedded program object allows a user to view large and  
complex multi-dimensional objects from within the browser's  
window. The user can manipulate a control panel to change the  
20 viewpoint used to view the image. The invention allows a  
program to execute on a remote server or other computers to  
calculate the viewing transformations and send frame data to  
the client computer thus providing the user of the client  
computer with interactive features and allowing the user to  
25 have access to greater computing power than may be available  
at the user's client computer.

104-101-4444-0000

**DECLARATION**

As a below named inventor, I declare that:

My residence, post office address and citizenship are as stated below next to my name; I believe I am the original, first and sole inventor (if only one name is listed below) or an original, first and joint inventor (if plural inventors are named below) of the subject matter which is claimed and for which a patent is sought on the invention entitled: **EMBEDDED PROGRAM OBJECTS IN DISTRIBUTED HYPERMEDIA SYSTEMS** the specification of which   X   is attached hereto or        was filed on        as Application Serial No.        and was amended on        (if applicable).

I have reviewed and understand the contents of the above identified specification, including the claims, as amended by any amendment referred to above. I acknowledge the duty to disclose information which is material to the examination of this application in accordance with Title 37, Code of Federal Regulations, Section 1.56. I claim foreign priority benefits under Title 35, United States Code, Section 119 of any foreign application(s) for patent or inventor's certificate listed below and have also identified below any foreign application for patent or inventor's certificate having a filing date before that of the application on which priority is claimed.

**Prior Foreign Application(s)**

Country	Application No.	Date of Filing	Priority Claimed Under 35 USC 119
			Yes <u>      </u> No <u>      </u>
			Yes <u>      </u> No <u>      </u>

I claim the benefit under Title 35, United States Code, Section 120 of any United States application(s) listed below and, insofar as the subject matter of each of the claims of this application is not disclosed in the prior United States application in the manner provided by the first paragraph of Title 35, United States Code, section 112, I acknowledge the duty to disclose material information as defined in Title 37, Code of Federal Regulations, section 1.56 which occurred between the filing date of the prior application and the national or PCT international filing date of this application:

Application Serial No.	Date of Filing	Status
		<u>      </u> Patented <u>      </u> Pending <u>      </u> Abandoned
		<u>      </u> Patented <u>      </u> Pending <u>      </u> Abandoned

Full Name of Inventor 1	Last Name Doyle	First Name Michael	Middle Name or Initial	
Residence & Citizenship	City Alameda	State/Foreign Country California	Country of Citizenship USA	
Post Office Address	Post Office Address 5 Rimmel Court	City Alameda	State/Country California	Zip Code 94502
Full Name of Inventor 2	Last Name Martin	First Name David	Middle Name or Initial	
Residence & Citizenship	City San Jose	State/Foreign Country California	Country of Citizenship USA	
Post Office Address	Post Office Address 5572 Makati Circle	City San Jose	State/Country California	Zip Code 95123
Full Name of Inventor 3	Last Name Ang	First Name Cheong	Middle Name or Initial	
Residence & Citizenship	City Pacifica	State/Foreign Country California	Country of Citizenship Malaysia	
Post Office Address	Post Office Address 658 Hickey Blvd.	City Pacifica	State/Country Pacifica	Zip Code 94044

I further declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under section 1001 of Title 18 of the United States Code, and that such willful false statements may jeopardize the validity of the application or any patent issuing thereon.

Signature of Inventor 1	Signature of Inventor 2	Signature of Inventor 3
Michael Doyle	David Martin	Cheong Ang
Date	Date	Date